

(19) World Intellectual Property Organization  
International Bureau



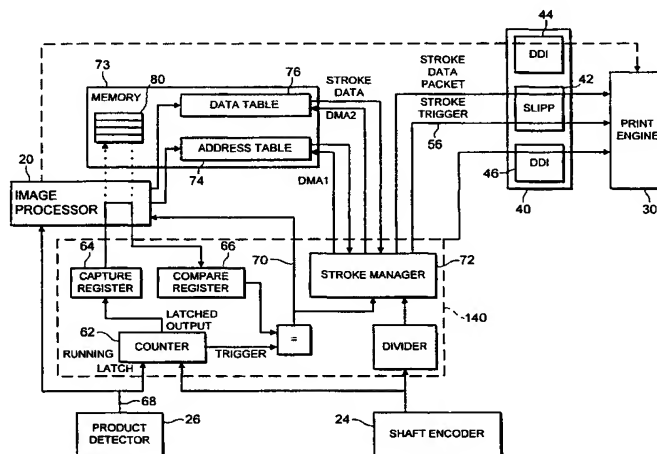
(43) International Publication Date  
22 February 2001 (22.02.2001)

PCT

(10) International Publication Number  
**WO 01/13328 A1**

- (51) International Patent Classification<sup>7</sup>: **G06K 15/16**, G03G 15/00
- (74) Agent: **MCGOWAN, Nigel, George**; Marconi Intellectual Property, Waterhouse Lane, Chelmsford, Essex CM1 2QX (GB).
- (21) International Application Number: PCT/GB00/03173
- (22) International Filing Date: 16 August 2000 (16.08.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
09/375,100 16 August 1999 (16.08.1999) US
- (71) Applicant: **MARCONI CORPORATION PLC**.  
[GB/GB]; One Bruton Street, London W1X 8AQ (GB).
- (72) Inventors: **STAMER, Michael, E.**; 6648 N. Avers, Lincolnwood, IL 60645 (US). **KLEIN, Michael, A.**; 1093 King Charles Court, Palatine, IL 60067 (US). **POTZLER, J., Edward**; 3519 Prairie Avenue, Brookfield, IL 60513 (US). **PICKELL, James, Robert**; 1191 Independence, Bartlett, IL 60103 (US). **EMERSON, John, Keightly**; 424 Franklin Boulevard, Elgin, IL 60121-4439 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— With international search report.
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: PRINT ENGINE CONTROL SYSTEM



(57) Abstract: A control system capable of being connected to a plurality of printers (30), said printers (30) employing print engines (30) of the same or different configurations, and using the same or different data protocols, each printer (30) including a print head controlled by the associated print engine (30) to mark a substrate, at least one of said print head and said substrate moving relative to the other at a rate which is not controlled by said control system or said printer (30), said control system comprising: a) an image processor (20) capable of generating image data in more than one protocol representing individual pixels or vectors to be marked on said substrate; b) a synchronization signal generator (24, 72) for generating synchronization signals representative of the relative location of said print head and said substrate; and c) an interface connecting said image processor (20) and said synchronization signal generator (24, 72) to said print engine (30), and allowing said image data and synchronization signals to be transferred to said print engine (30), said interface accommodating different print engine protocols, thereby to allow various print engines (30) to be interfaced to the same control system.

PRINT ENGINE CONTROL SYSTEM

The present invention relates to printers, and in particular to a method of creating printer systems using one or more print engines which can be used in conjunction with a variety of other devices.

5 Industrial marking printers typically include a production line on which items to be marked move along a conveyor system and are detected by a product detector in the vicinity of a print-head. In some cases, a shaft encoder tracks conveyor movement. A shaft encoder generally comprises a shaft and a detecting/encoding assembly which detects the rate of rotation of the shaft and outputs pulses with a period representative of the rate of rotation of the shaft. The shaft  
10 encoder is placed in mechanical communication with part of the conveyor system which moves at a speed proportional to the rate at which the product will move past the print-head, and generates pulses at some specific rate of pulses per inch of travel; the higher the velocity of the conveyor, the higher the frequency of the pulses will be. The pulse interval is called the encoder pitch, expressed in pulses per inch.

15

If, for example, the printer is a continuous ink jet printer, it will generally be programmed to direct droplets of ink toward a substrate comprising part of a product, whereby to print text or images on the substrate as it passes the printer's print head. The timing of these droplets is critical for printing legibly at the right location on the substrate.

20

As the rate of movement of the production line is not fixed, the timing of the droplets directed to the substrate must be controlled by the position of the production line, and hence the output of the shaft encoder is used to time the droplets. Often, printers are designed to print a set

of droplets representing a column of dots in quick succession, referred to as a stroke. The time interval between droplets making up a stroke is often independent of the speed of the production line. Shaft encoder pulses are accordingly used to trigger strokes in such printers. Often, the raw encoder pitch is higher than the planned print stroke pitch, so the encoder pulse rate must be  
5 divided by some amount. Usually this is an integer value such as eight, ten, etc.

Product detectors are devices which sense the presence of an object and produce either an active high or an active low output signal. They are used to synchronize printing of text or images with the passing products on the production line, as the products are not guaranteed to be  
10 equi-spaced on the production line.

Software running on the print engine processor controls accessing of the image data and ensuring that appropriately charged droplets representing the image data are output at the appropriate time.

15 A counter also running in software on the print engine processor is triggered by an interrupt every time a divided pulse is output from the shaft encoder. The print engine software is also interrupted by a product detect trigger pulse, at which time the present value of the counter is stored. The software tracks the value of the counter, and when it reaches a specified value, a  
20 fixed number greater than the value when the product detect trigger occurs, commences printing of a stored image or text on the substrate. The counter delay will depend on the relative positions of the product detector and the print head. This type of operation is inefficient because the processor controlling the image printing is continually interrupted by the shaft encoder pulses to update the counter, and during the interrupt is not able to process any image data.

In certain cases, where the production line speed is relatively constant, the shaft encoder expense may be eliminated by substituting an internal timer which generates pulses at a rate suitable for emulation of an encoder pulse rate. Usually, this internal timer is programmable through software control, to be set to an appropriate pulse rate for the particular line speed being  
5 used at the time of printing. This internal pulse rate is then used to trigger printed strokes just as with an external shaft encoder. The only side effect of internal timing instead of an external encoder is that if the actual line speed varies from the planned speed, the printed image will be either stretched or compressed, depending on whether the speed increases or decreases.

10 Dividers are known which divide the encoder pulse rate by a non-integer value which is actually a ratio of two integer values. Such a divider was utilized in the Cheshire 4000 Imaging System, produced by the present assignee.

15 Stroking systems based around the use Direct Memory Access (DMA) to access image data are well known. For example, a printer system of the present assignee known as Apollo2 has a CPU bus interfaced to an Altera 9560 Erasable Programmable Logic Device (EPLD). The EPLD interfaces directly to an SRAM device. All CPU reads and writes are handled through the EPLD, although this is transparent to the CPU.

20 The CPU builds the image to be printed in the SRAM device. When a message is to be printed, the CPU flags this to the EPLD. At this point, the EPLD transparently reads parameter values from the SRAM, including the start and end stroke addresses and the stroke interval, and state machines within the EPLD read stroke data from the SRAM at the required intervals. Once the stroke has been read from the SRAM, it is serially shifted out to a serial to parallel convertor  
25 located at the print head. Strokes are limited to a single word in length.

On every bus cycle, the EPLD decrements timer variables to ascertain the time at which the next stroke is to be printed. The address of the stroke to serialize is stored in the parameter memory location. This value is incremented by the EPLD when the stroke has been serialized, and compared with the end address as specified by the CPU. If the two match, the state machines reset and the 9560 interrupts the CPU.

Many known continuous ink jet printers transmit messages to print heads using a character coding system, which does not lead to any control of images at a finer detail than predefined characters. The printer is effectively limited to fonts programmed into the print head. Strings of characters might be single lines or may be arranged as multiple lines of characters. In each these types of systems, the print image is communicated to the printer with character codes, such as ASCII codes. The character code is a bit pattern of only a few bits, usually 7 or 8, representing the entire character. Character based printing can of course be used with dot matrix printing systems; however, the pattern of dots in each character is predefined.

When a dot matrix printing system is being used, a print image can be addressable as an array of dots or pixels rather than just a string or array of characters. This is a more generalized and flexible concept for a printing system. With such a system, each character can be rendered into the intended array of matrix dots, some of which are to be printed, and others left blank. A character rendered in a particular font is represented by an array of pixels. The font chosen will usually have an associated height of some number of pixels, and the width may be fixed or variable. However, the font can easily be changed, even within the same message, without any change to the format of data that the print head can accept. The character images may be placed wherever intended within a planned total bit map image area. Lso, other image components

which are not characters at all ,but are actual graphical shapes, such as logos and “pictures” may be placed as needed to construct the total bit map representation of the intended print image.

Likewise, many products are available which print using a steered beam of  
5 electromagnetic radiation, or other vector driving systems such as ink plotters. Such systems are generally addressed using a vector based addressing scheme, moving the beam or plotting mechanism between points on a substrate, either in straight lines or other definable paths. Such vector printing schemes, for the purposes of this patent application, are very much akin to “sub  
10 character” level dot matrix printing, as they allow the printer to print images or text in almost any pattern within the physical limits of the printing mechanism. These are the primary types of printing mechanisms to which the invention is addressed. The actual method a print head uses to mark a substrate is not the primary concern of the invention, dot matrix, drop on demand or other marking mechanism could be used.

15 According to a first aspect of the present invention there is provided a control system capable of being connected to a plurality of printers, said printers employing print engines of the same or different configurations, and using the same or different data protocols, each printer including a print head controlled by the associated print engine to mark a substrate, at least one of  
20 said print head and said substrate moving relative to the other at a rate which is not controlled by said control system or said printer, said control system comprising: a) an image processor capable of generating image data in more than one protocol representing individual pixels or vectors to be marked on said substrate; b) a synchronization signal generator for generating synchronization signals representative of the relative location of said print head and said substrate; and c) an interface connecting said image processor and said synchronization signal generator to said print  
25 engine, and allowing said image data and said synchronization signals to be transferred to said

print engine, said interface accommodating different print engine protocols, thereby to allow various print engines to be interfaced to the same control system.

According to a second aspect of the present invention there is provided a control system  
5 for a printer having a print engine and a print head controlled by said print engine for marking a moving substrate, said control system comprising: a) an image processor for generating images to be output to said print engine, for causing said print head to mark said substrate; b) a counter operating independently of said image processor and responsive to signals generated by a shaft encoder associated with movement of said substrate; c) means for reading the value of said  
10 counter, adding a predetermined value to said counter value and storing the result as a compare value in response to the detection of said substrate; and d) comparing means operating independently of said image processor for initiating printing of said images on said substrate when the value of said counter matches said compare value.

15 According to a third aspect of the present invention there is provided an interface for connecting a control system to one of a plurality of printers, said printers employing print engines of the same or different configurations, and using the same or different data protocols, each printer including a print head controlled by the associated print engine to mark a substrate, at least one of said print head and said substrate moving relative to the other at a rate which is not  
20 controlled by said control system or said printer, said interface comprising: a) a stroke signal interface capable of transferring stroke data in more than one protocol representing pixels in said stroke from said control system to said print engine; and b) a synchronization signal interface for transferring synchronization data representing the position of said substrate relative to said print head from said control system to said print engine.

The present invention provides a printer system for printing on a substrate moving relative to a print head at a rate which is beyond the control of the printer system. The print head might be stationary as the substrate moves past, or the print head might move while the substrate remains stationary. Alternatively, both substrate and print head could both move independently during operation. While the specific embodiments of the invention described relate to continuous ink jet printing, the invention is also applicable to other printing devices in which substrate and print head move relative to one another at a rate which is not under the control of the printing system. For example, any dot matrix based system allowing pixel level addressability would be within the scope of the invention. A steered beam laser device using a vector based, or even pixel based, addressing mechanism would also be within the scope of the invention.

The present invention provides a continuous ink jet printer system wherein droplet generation is carried out by a print engine connected across an interface to apparatus which generates image data and timing signals. Image data is transmitted across the interface and timing signals transmitted across the interface trigger the printing of the image data.

The present invention further provides a method of counting shaft encoder pulses and generating stroke trigger pulses therefrom. This specific aspect of the invention provides a method for dividing the frequency of shaft encoder pulses by a non-integer number to generate stroke pulses. According to a specific embodiment, the non-integer ratio may be any number expressible as a ratio of two eight bit numbers. With the divide ratio set, the print stroke rate or pitch is the encoder rate divided by  $N$ , where  $N$  is the divisor number. This allows much more accurate timing of strokes and allows fonts to be correctly proportioned when the stroke dot pitch in the direction of movement of the substrate is not a multiple of the distance moved by the



production line between encoder pulses. The use of a two integer ratio is particularly useful as the relative values of the dot pitch and the distance moved between encoder pulses is often closely related to gearing ratios on a production line.

5           The invention also provides a hardware based encoder pulse counter mechanism, referred to hereinafter as a capture and compare mechanism. The capture and compare system comprises a free-running counter which continuously counts encoder pulses. These pulses could be divided or undivided pulses, but in the presently preferred implementation are undivided pulses. Preferably, this counter does not execute on a processor which carries out image processing or  
10   stroke generation, and ideally the counter will execute on hardware dedicated to the synchronizing of strokes. This counter is interrupted by product detect events and the counter value stored therein when the product detect occurs is stored. The output value of the counter is monitored and when the counter reaches a predetermined value relative to the stored counter value, image data is generated.

15

          In a particular embodiment, when a product detect trigger pulse occurs, the present value of a counter is captured and an interrupt to an image processor is generated. The interrupt routine reads the captured value, adds to this the amount of print delay desired, and then sets the resultant computed value into a compare register. The routine also initiates the setup of the next  
20   print image, memory pointers etc. A second interrupt will be triggered when a match is detected by a dedicated comparing circuit between the encoder counter and the compare value. This second interrupt pulse initiates a stroke manager system to begin the stroke printing process. This process consists of a primary Direct Memory Access (DMA) hardware obtaining addresses of strokes in memory from a list in memory and a secondary DMA obtaining stroke data from the  
25   addresses obtained by the primary DMA and sending stroke data to a print engine at a rate

determined by a shaft encoder. Stroke printing generally continues without processor intervention until message printing is completed.

The invention further provides means for aligning the images of several print heads very  
5 precisely and under software control so that these images can be combined into a single, larger,  
multiple head image. This means includes the software controlled print delay and encoder divide  
logic, a set being provided for each head. With this hardware, the print position and alignment  
between heads can be controlled to a resolution which is a fraction of a stroke interval - the  
undivided encoder interval.

10 When line speed is sufficiently constant that operation without the use of a shaft encoder  
is preferred, an internal timing signal may be generated to take the place of the shaft encoder  
pulses. Here, an internal pulse rate can be generated which either emulates the desired stroke rate  
or the undivided encoder rate. In the latter case, the undivided, simulated encoder rate is then  
15 further divided in the same way as with an external encoder.

The invention will now be described, by way of example, with reference to the  
accompanying drawings, in which:-

20 FIGURE 1 shows an overview of a printer system embodying two sets of hardware in  
accordance with the invention. Each set of hardware operates two print heads;

FIGURE 2 shows a detailed view of one of the sets of hardware of Figure 1. To make the  
figure more clear, only the hardware required to operate one of the two print heads is shown;

FIGURES 3 and 4 show a representation of a production line in which a product detector and a print head are separated by different distances;

FIGURE 5 shows values of a counter in an implementation of a stroke pulse manager which outputs pulses at a non-integer ratio of the pulses output by a shaft encoder;

FIGURE 6 shows a first image generated by two print heads of the invention;

FIGURE 7 shows a second image generated by two print heads of the invention;

FIGURE 8 shows in more detail a stroke level interface shown in FIGURE 1;

FIGURE 9 shows a data word layout for data transmitted over the stroke level interface shown in FIGURE 1;

FIGURE 10 shows the relationship between serial data interface signals provided over the stroke level interface shown in FIGURE 2;

FIGURE 11 shows stroke trigger signal timing over the stroke level interface shown in FIGURE 8;

FIGURES 12 and 13 show single buffering and double buffering stroke data timing signaling respectively;

FIGURE 14 shows the DDI interface of FIGURE 1 in more detail; and

FIGURE 15 shows the reset interface of FIGURE 1 in more detail.

Figure 1 shows an overview of a continuous ink jet printer 10.

5

The following terms are defined in order to enable consistent interpretation of this document. The control system is the portion of a complete printer that communicates with and controls print engine(s) 30; in essence, this is the rest of the printer system external to the print engine(s). The image processor (IP) is a portion of the control system generating the bit map of the image to be printed and sends stroke data and trigger signals to the print engines. The print engine 30, in this embodiment, is a technology specific printing mechanism capable of printing columns of dots (strokes) in response to stroke data and trigger signals. The stroke data of the particular embodiment described can represent any arrangement of dots that the printer is capable of printing. Bit maps representing the images to be printed are stored in memory, and strokes representing rows or columns of the bitmap image are output to the printhead. However, there is no reason why the print engine could not use a different technology, such as a steered beam. The print engine may have one head or more. In this embodiment, only one trigger is provided, so all heads in the print engine respond to the same stroke trigger signal. When multiple triggers are needed, multiple triggering interfaces can be used. A product detect signal (referred to as Product Detect) is a signal generated by a product detect device. The signal indicates that a product has entered a printing region. A print request signal (referred to as Print Request) is an internal signal generated by the image processor which indicates that a product or substrate piece has progressed to a point where printing is to begin. The Print Delay is the difference, measured in substrate travel, between the product detect point and the print request point.

10

15

20

25

One or more print engines 30 driving associated print-heads 12 are interchangeably connected via print engine interfaces 40 to controller hardware 14. These interfaces facilitate reuse of the majority of the control system for such printers by meeting the needs of several printer products which have a similar range of stroke data rates. A complete set of Stroke  
5 Manager hardware 140 is provided for each print engine which the controller hardware 14 operates. The essential components of each set of stroke manager hardware is shown in Figure 2.

The controller hardware, and in particular the image processor, receives signals from up to two product detectors 26 and up to two shaft encoders 24, although the number of shaft  
10 encoders and product detectors could easily be varied and still remain within the scope of the invention. Generally, each shaft encoder and product detector is associated with one of the print engines. While the embodiment described herein operates only two print engines, there is no reason why functionality for more than two print engines could not be incorporated. More complicated dependencies between the product detectors, shaft encoders and print engines could  
15 be made available within the hardware and still be within the scope of the invention.

An image processor 20 communicates with the controller hardware 14 and generates image data to be output to the print engines which is output to memory 73. The image processor of this embodiment is preferably implemented using a Motorola 68322 processor. It receives  
20 communication from the controller hardware via its two interrupts and inputs and outputs data to registers in hardware. The image processor also communicates with other devices to set up print images. However, the details of image generation within the image processor are beyond the scope of this application and will not be discussed further herein.

The controller hardware functions are hereinafter described in more detail with reference to Figure 2 so that it can be understood how the print engine 30 associated with each print head 12 is synchronized with the substrate on which printing occurs. The hardware functions are implemented using programmable logic circuits. Figure 2 shows stroke manager hardware 140 which receives input from a single product detector and a single shaft encoder and outputs to a single print engine. The operation of the stroke manager hardware is duplicated for each print engine which each set of controller hardware can handle.

This embodiment of the invention uses a principle referred to hereinafter as “capture and compare” to synchronize printing with the passing substrate on the production line. This results in images which can be printed at significantly delayed positions relative to the head location at the time of product detect.

The core of the capture and compare system is a free-running counter 62 running on programmable hardware which continuously counts undivided encoder pulses from the shaft encoder 24. In this embodiment the counter is a 16-bit counter, although clearly the number of bits used could be varied appropriately depending on the pulse rate of the shaft encoder and the distance between products. Assuming that the print engine is ready and printing is enabled, the sequence operates as follows. The product detector 26 senses the product to be marked. The selected product detect edge pulse triggers an interrupt 68 to inform the image processor 20 of the product detect event. This edge pulse also triggers the capture, or latching, of the contents of the free-running encoder pulse counter 62 into capture register 64. Its relative value is used to control the print request event. During the interrupt routine on the image processor 20, the captured value is read from the capture register 64 by the image processor, and the amount of print delay desired in undivided encoder ticks is added to the value by the image processor. This

resultant computed value is written from the image processor into the compare register 66. The routine also initiates the setup of the next print image, memory pointers, etc. by the image processor 20. It is to be appreciated that the following function performed by processor 20 could instead be performed by an additional means separate to processor 20: reading the value of capture register 64; adding the print delay to it; and storing the result in compare register 66.

A second interrupt of the image processor, referred to as the print request interrupt 70, is triggered at some later point in the encoder travel, when a match is detected between the value in the compare register 66 and the value in the counter 62. A dedicated comparing circuit 71 monitors the output of the compare register 66 and the counter 62 and generates this interrupt. This interrupt pulse also initiates the hardware stroke manager 72 system to begin the stroke printing process. The stroke manager contains DMA hardware which directly accesses memory 73 to acquire stroke data and send it to the print engine at each divided encoder interval. More specifically, a primary DMA channel accesses an address table 74 and feeds a secondary DMA channel with addresses for the stroke data in the data table 76. This allows much more freedom in data layout than is possible if the stroke data must all be laid out sequentially. Stroke printing generally continues without processor intervention until message printing is completed.

As the monitoring of the product detect and print request events is carried out using dedicated hardware, a significant amount of overhead is saved by the image processor 20 where monitoring of shaft encoder pulses has heretofore been carried out. Monitoring by the image processor requires interrupt routines being executed therein every time there is shaft encoder pulse. This uses a significant number of processor cycles.

A situation is now considered in which the distance  $x$  between the product detector 26 and print head 12 is more than the distance  $y$  between the position of a product detector and the position on the previous product at which printing is to commence, as shown in Figure 3.

5 In this situation, if a product detect occurs, the product detect time would need to be stored by the image processor, rather than being adjusted to calculate a print request time and written straight to the compare register. Once a print request interrupt 70 occurs, the interrupt routine on the image processor would then write the print request time to the capture register, as the value in the print request is no longer needed.

10 A situation will now be considered with reference to Figure 4, in which the distance  $x$  is greater than the distance  $z$  between the print position on one product and the product detect position on the next but one product. At the instant shown, the print request time for product P1 will need to be stored in the capture register, the print request time (or the product detect time) of  
15 product P2 will need to be stored in the image processor and the print request time (or the product detect time) of product P3 will also need to be stored in the image processor.

A way to overcome these problems and provide a general purpose system is to provide a queue 80 of print request times stored in memory 73. Every time a product detect interrupt  
20 occurs, the product detect counter value, or the calculated print request counter value is added to the queue. If no value has been written to the compare register since the print job was commenced, the print request counter value calculated can be written straight out to the compare register. Whenever a print request interrupt 70 occurs, the next print request counter value in the queue (or the next product detect value in the queue with the appropriate print delay added if this  
25 implementation is used), is written into the compare register 66. This allows appropriate printing



to occur whatever the distance between the products, the product detector and the print head. Furthermore, the queue is only accessed or written to in response to product detect or print request interrupts and there is therefore very low overhead on the image processor.

5           The stroke manager hardware 72 is responsible for sending stroke data packets to the print engine, or print engines in embodiments with more than one print engine. This happens provided that various hardware registers are set up properly ahead of the print request. Those registers and the operation of the stroke manager hardware are hereinafter described for reference.

10           To summarize, the stroke manager is a hardware direct memory access (DMA) system that is custom designed to accomplish the task of sending stroke data generated in memory 73 by image processor 20 to the print engines 30. The sending of the stroke data is coordinated using stroke trigger signals 56 which in this embodiment consist of a train of pulses for each print  
15   engine coordinated with the signals from the associated shaft encoder 24, as will be discussed.

Several hardware registers are used to assist in the operation of the stroke manager. Some need only be set up once or until the job configuration requires a change. For example, the stroke height is constant and need not be changed unless the image height changes. Other  
20   registers must be initialized prior to each print request.

The image processor 20 prearranges two data areas 74, 76 for a print image. These are:

1. A table 74 of stroke data starting addresses.

2. The stroke data itself 76 located at the locations pointed to by the table.

The DMA process involves two steps. Firstly, the address of a stroke of data is fetched by a primary DMA channel from a stroke address table 74 prepared by the image processor 20 along with the actual stroke data pattern 76. Second, the data bytes representing the stroke are fetched via a secondary DMA channel using the address information obtained from the primary DMA access and the complete stroke data packet representing the stroke is sent to the print engine 30.

This process is designed so that the stroke manager software can be totally flexible in its use of image data, under the control of the image processor which sets the appropriate registers used by the stroke manager. The stroke manager can reuse an image repeatedly by just setting the table pointer to the same image over and over again. It can reuse just a portion of an image while surrounding the fixed portion with variable information that changes with each new print image, thus saving processing time wherever reuse makes sense.

The stroke trigger signal 56 originates from either an internal timer in the image processor or a shaft encoder divide channel. A shaft encoder divide channel is a pulse rate divider 25 which will take a quadrature output from a shaft encoder 24, multiply the rate by four, and then allow divide down by an integer or non-integer ratio within an allowed range.

According to this embodiment the non-integer ratio  $N$  may be any number expressible as a ratio of two eight bit numbers. A stroke trigger signal is originated for every  $N$  shaft encoder pulses. This is easily accomplished, as shown in Figure 5, by incrementing a counter by the lower of the two integer values each time an encoder pulse arrives. When the counter value

reaches the higher of the two integer values, a stroke trigger signal is generated, and the counter value is decreased by the higher of the two integer values. For example, if the ratio was  $15/2$ , for each shaft encoder pulse the counter would be incremented by 2, until, after 8 pulses, the counter value would reach 16, above the larger integer value of 15. A stroke signal would then be generated and the counter value would be decreased by 15 to a new value of 1. Now, after 7 pulses the counter would again reach 15. Another stroke pulse would be generated and the value would again be decreased, this time to 0. This cycle would repeat and stroke pulses would be generated alternately every 7 or 8 encoder pulses. This is a best approximation to the required strokes every 7.5 encoder pulses, and importantly does not stray away from the desired rate over time.

As different stroke managing hardware is used for each print head, each print engine can have a different print delay after a product detect. Different print heads might respond to the same product or different product detects. Furthermore, as the print delay is based on the undivided shaft encoder pitch, the point at which printing commences on a substrate can be controlled to a resolution of a fraction of a stroke. Likewise, as each print head utilizes a different divider, subsequent strokes subsequent stroke triggers for each print head are based on the initial stroke time, with fraction of a stroke resolution. Finally, print queuing of several print images is possible for each engine, so that different print heads can be printing images for different print images at substantially spaced distances along the production line. However, with the fraction of a stroke resolution, the different print heads can be aligned to print different parts of the image, and the images can be "stitched" so that there is no visible interface between adjacent images. Examples of such stitching are shown in Figures 6 and 7. The line of interface between the images generated by two print heads is shown by a dotted line. In the first example, shown in Figure 6, multiple lines of text are provided. Head to head synchronization of

the content is critical, but pixel level alignment is not of the utmost importance for readability and good appearance. In the second example, shown in Figure 7, large components are generated as a multiple head image. The head to head synchronization is critical, but the relative position of each head's image position is also now critical. Ideally, placement should be performed with sub-stroke interval accuracy approaching a seamless transition between heads in the image.

The design of the print engine interface 40 of the presently preferred embodiment is hereinafter described in more detail, with reference to Figures 1, 2 and 8-15. It consists of three parts, as shown in Figure 2:

- The stroke level interface, 42
- the device driver interface (DDI) 44
- additional auxiliary signals 46

The stroke level interface 42 sends stroke data to the print engine and triggers the printing of each stroke. The DDI 44 is a full duplex serial port which allows translated Virtual Control Network (VCN) dialog to take place between the image processor and the print engine. The added signals contain a hardware reset signal.

The stroke level interface 42, shown in detail in Figure 8, provides real time communications between the image processor 20 and print engines 30 connected thereto. Its function is to provide stroke triggers and stroke data to the print engines 30 in real time. The stroke trigger is a signal that determines when the next stroke of an image should be printed. Stroke data is the stroke pattern for the next stroke. The normal sequence is that the data for a following stroke would be sent right after the trigger for a current stroke. Any new stroke data

will overwrite stroke data that is currently in the data buffers. This ensures stroke synchronization. If the print engine receives new stroke data without receiving a trigger, it flags this as an error.

5           The stroke level interface 42 would normally use Point-To-Point or Multi-Point implementations. The Point-To-Point SLI (SLIPP) has its performance and interface targeted at systems with one image processor 20 and one print engine 30. Systems that are implemented on one PCB board would use the Point-To-Point SLI. The SLIPP 42 can be made available to OEMs. The Multi-Point SLI 42 has its performance and interface targeted at systems that have  
10       one image processor 20 that is responsible for multiple print engines 30.

          There are three levels to the interface 42 - the data frame 420, the parallel port 422 and the Serial Port 424. The data frame 420 is a software level. The parallel 422 and serial ports 424 are hardware levels. If the complete printer system were to be a single board printer with no  
15       expansion to drive additional boards, then the interface would use only data frame 420 and parallel port level 422.

          The image processor and the associated controller hardware could be a significant distance from the print engines, and this distance is highly variable. A serial interface is therefore  
20       used to connect the image processor 20 to separate print engine 30 or print head 12 boards. A serial interface 424 is used to keep the number of cable wires to the minimum necessary to produce a reliable, low EMI system.

          Stroke data is transferred as a packet of data words, as shown in Figure 9. The stroke  
25       data word comprises a command/data flag and eight data bits. The choice of nine bits for the

words is based on the same use in larger multi-point systems using the hot-link interface chip. The nine bit word uses the most significant bit as a Command/Data flag followed by eight data bits.

5           Stroke data packets consist of a series of data words - start byte, Head Number, Product ID, discretionary byte(s), stroke print image bytes and end byte. The presently preferred format is shown in Figure 9. The Cypress HOTLINK® chipset is the currently preferred means for serializing/deserializing the data.

10           The serial interface 424 consists of three signals which form a reliable serial interface. The three signals are:

- Serial Stroke Data
- Serial Clock
- 15       - /Frame\_Sync

Figure 10 indicates the basic relationship between the three serial data interface signals. These signals take the physical form of differential pairs. The driver end is a low voltage differential driver which swings about 1/3 volt peak to peak into a 100 ohm load. A typical  
20       differential driver is the National Semiconductor DS90C031.

At the print engine 30, line terminating impedances and differential receivers are required. The matching receiver is the National Semiconductor DS90C032. The manufacturer's application notes provide suggested termination alternatives.

In order to support the print engine data rates required, the data rate is selectable. The maximum data rate intended is 32Mbits/sec. Binary sub-multiples of 16, 8 and 4 Mbits/sec. may also be chosen. Clearly other data rates could be used depending on the implementation. The top rate used in this embodiment is based on the need to exceed a projected 128 bit/head, dual  
5 headed system at 50K strokes per second. The idea of allowing lower data rates accommodates lower technology print engines as well as reduced EMI where the higher data rate is not needed.

The serial clock signal is continuously present at the interface to facilitate synchronizing nozzle frequencies across multiple print engines if needed for improved pixel stitching. Stroke  
10 data words are clocked in the order of least significant bit first, followed by the remaining seven data bits, the Command/Data flag bit and then the parity bit. The Frame Sync signal goes true at the start of the first data bit cell, roughly coincident with the high to low transition of the serial clock signal.

15 The parity bit is set to a "1" when the number of 1s in the command bit plus data bits is even and "0" when the number of 1s is odd. Accordingly, as there are an odd number of data bits per frame, the serial data signal should never contain all 1s or all 0s within a single frame.

Note that the logical polarity assigned to the interface driver for the Frame Sync signal is  
20 active low. This sense was chosen because the differential receiver output will pull high when the input is shorted or open. Therefore, when the Print Engine Interface cable is disconnected, the Frame Sync signal is held inactive. Figure 8 shows /FRAME\_SYNC as the signal designation at the interface receiver.

The data transmitted over the SLIPP 42 will now be discussed. The print engine 30 receives vertical strokes and stroke triggers from the image processor 20 and then uses technology specific techniques to mark the substrate. The print engine 30 is responsible for maintaining any technology specific functionality that may be related to the type of printer involved.

Using the defined SLI 42, the print engine 30 accepts a vertical stroke and then prints that stroke when it receives the stroke trigger.

Although the currently preferred design has been focused on matrix based printing, there is no reason why the design could not be modified to handle vectored information. This would be required for example to drive a steered beam laser product.

The following are error conditions which can occur in the system of the present embodiment of the invention:

Print Overspeed - The condition that would occur if a stroke trigger for a new stroke is sent to the print engine 30 before the print engine completed the previous stroke.

Data Overspeed - The condition that would occur if a SLI data word was sent, but the print engine data buffer was not ready to receive it.



Stroke Data Error - The condition that would occur if a print engine received a trigger for a new stroke before receiving a complete new data packet including the End of Frame code.

- 5    Lost Trigger Error - The condition that would occur if a new stroke data packet is fully received before a trigger is received for the previous stroke.

Each packet of stroke data must be sent sufficiently ahead of the stroke trigger signal to allow the print engine 30 to process the information. This may take more time for some print  
10 technologies than others. Also, the transmit time for the stroke data to all heads 12 in a printer must be well within the expected minimum stroke time interval.

Stroke data may be single buffered, double buffered or multiple buffered if needed for the print technology involved. The buffering must remain fixed so as to produce a known delay.  
15 Then the control software and print engine software must both be made aware of the buffering scheme used.

The sequence of events is as follows. Timing diagrams for buffering schemes described below are shown in Figures 11, 12 and 13. After a product detect occurs, a print delay is counted  
20 out. Normally, this would be a count of undivided stroke pulses. At that point, a print request occurs, the encoder divide logic is cleared and the encoder divide intervals begin. The first stroke packet may be sent during or after the print delay; it may also be sent immediately after the occurrence of the product detect since the print delay can be set to zero. Each subsequent stroke trigger will then enable the sending of the next stroke data until the final stroke of the print  
25 image has been sent and triggered.

The start byte of each packet defines whether the stroke is the first, middle or last stroke of the image. This allows the print engine to perform any special activation sequence if needed at the start and also allows it to schedule any “print idle” functions between print cycles.

5 All packet data words are expected to be received by the print engine. There are no status signals provided which would indicate that an engine 30 was not ready to receive. In the event that data is sent when the print engine 30 is not ready to receive it, the print engine must detect this condition in order to communicate to the image processor 20 through the DDI 44 dialog that a data overspeed has occurred.

10 If the print engine 30 does not need to buffer more than one stroke of print data, the stroke data and trigger signals would exhibit the relationship shown in Figure 12. Figure 11 shows the basic relationship between product detect, print delay and print request.

15 If the print engine 30 needs more time to process strokes, then double buffering may be used. With double buffering, a stroke data packet is not printed until the 2nd stroke trigger following the data. Both the control system and the print engine software need to be aware of this. Figure 13 shows the timing for double buffering.

20 The sequence of events would be as follows. The first stroke data packet in a print image is sent after the print request as before. Then, the stroke trigger will serve to provide an interval before a stroke trigger signal is generated for sending the next stroke data packet. Once the 2nd packet is sent, the following trigger causes the first packet to be printed. Subsequently, the 3rd packet is sent. The next trigger causes the 2nd packet to be printed. The process continues until  
25 the end of the image. At this point, the sequence becomes different than for single buffering.

Since the buffer has two stages, the last print stroke will be stuck in the buffer unless another non-print stroke is sent. Therefore, the image has to have a trailing blank stroke. When this stroke is sent, its start byte is coded as a last stroke packet to signal this to the print engine 30. Receipt of this blank stroke causes the subsequent trigger to print the last real stroke.

5

By building the image so as to match the type of buffering used, the sequence of stroke data followed by trigger signal is kept universal.

Stroke data words are converted to serial form before sending over the SLIPP 42 cable.

10 In the process, a parity bit is added as a tenth serial data bit. This allows the print engine 30 to check parity before removing the bit at that end.

The stroke trigger signal is essentially a strobe pulse which will initiate the printing of a previously transmitted stroke pattern. The trigger signal is converted to a differential signal and  
15 transmitted over the identical type driver as used for stroke data packets. One differential trigger signal is grouped with each stroke data interface to form a complete Stroke Level Interface Point to Point (SLIPP) 42.

The stroke trigger signal consists of a short duration pulse, approximately 500 to 1000  
20 nanoseconds long. The lead edge of this pulse is defined as the trigger event. The duration was chosen so as to be relatively short, yet longer than several cycles of a print engine's logic clock, for synchronizing purposes.

The logical polarity assigned to the interface driver for the trigger signal is active low.  
25 This sense was chosen because the differential receiver output will pull high when the input is

shorted or open. Therefore, when the Print Engine Interface cable is disconnected, the trigger signal is held inactive. Figure 8 shows /TRIGGER as the signal designation at the interface receiver.

5           There are no status signals which could indicate to the image processor 20 that a print engine 30 is not ready to print a stroke when triggered. In the event that a stroke trigger is received before the print engine 30 is ready to respond to it, the print engine must detect this condition in order to communicate to the image processor 20 that this print overspeed condition has occurred.

10           A second interface to the print engine 30 is the device driver interface (DDI) 44, which connects the print engine to the virtual control network (VCN) in the preferred embodiment of the invention. A presently implemented embodiment does not have the print engine directly connected to the virtual control network, but still sends control signals through the DDI as a  
15           substitute for the VCN. The image processor 20 serves as the connection to the virtual control network for the print engines. The image processor translates VCN dialogs to communicate with each print engine. In a preferred embodiment the VCN uses a CORBA implementation. Using the DDI accordingly avoids the need for each print engine to run a real time operating system and CORBA Object Request Broker (ORB) at the DDI processor, thus eliminating those  
20           software burdens.

          The DDI interface 44 consists of a full duplex serial port as shown in Figure 14. The following are features of the presently preferred DDI:

- asynchronous data, allows connection to PC type serial ports using appropriate buffers;
- signals are differential pairs; Low Voltage Differential Signals (LVDS) as used on SLIPP 42;
- the baud rate is set by the print engine 30; the image processor adjusts to match;

5

The minimum baud rate used in the presently preferred embodiment is 9600 baud. The print engines 30 may use any baud rate equal to that or a multiple of 9600. The image processor 20 will find the correct baud rate to establish communications with the print engines 30.

10

Provisions are made for additional signals 46 within the Print Engine Interface 40. This includes a hardware Reset signal plus a spare signal in each direction.

15

20

The Reset signal is intended as a last resort, hardware reset input to the print engine 30. It would be generated by the image processor 20 after it had determined that the print engine 30 was not responding to a reset or other command via the DDI channel. The print engine will directly OR this input with its own internal power on reset in order to reinitialize its circuitry. A long time constant filter is recommended in this path to assure no chance of a noise transient causing a false trigger of the reset function. Therefore the image processor 20 will need to generate a sufficiently long reset pulse to overcome the filter delay. The presently preferred Print Engine Reset input responds to a 1 millisecond reset pulse width. The image processor 20 generates a minimum reset pulse width greater than 1 millisecond.

25

The physical form of the Reset signal as well as the spare signals is a low voltage differential signal (LVDS) as previously described. Note that the logical polarity assigned to the interface driver for the Reset signal is active low. This sense was chosen because the differential

receiver output pulls high when the input is shorted or open. Therefore, when the Print Engine Interface cable is disconnected, the Reset signal is held inactive. Figure 15 shows /RESET as the signal designation at the interface receiver.

5           According to the presently preferred embodiment, the Print Engine Interface 40 uses flat ribbon cables spanning as short a distance as possible in order to keep data reliability at a maximum and reduce EMI. A 26 pin latching ribbon cable header is presently preferred. Interleaved ground wire pairs between differential signal pairs are also used in an attempt to further reduce EMI and keep ground inductance down. At the same time, this plan allows for the  
10       use of twisted pair ribbon cable if an application requires it.

The following list of registers in controller hardware 14 comprise the controlling register set for the stroke manager. These are:

- 15           1.     Table address register - this is a 25 bit (for the 68322 processor) register which must be loaded prior to each print request with the address of the first entry in the stroke pointer table. As each stroke data packet is sent, this register is incremented automatically (or decremented for reverse direction) in hardware. Thus for each new print request, the start of the appropriate table must be  
20       reloaded even if it's the same as before.
2.     Data address register - this is a 25 bit register that is automatically loaded by the DMA hardware as it reads the contents of the stroke address table. This register is further incremented (or decremented) by hardware as the second phase of the  
25       DMA process fetches data words from successive memory locations.

3. Stroke Height register - this is a 9 bit register which holds the number of bytes in the stroke. Its contents only need be loaded once at the time that the initial stroke height is set. Its contents are copied into a down counter to count out the correct number of data bytes for the packet.

5

4. Stroke count register - this is a 16 bit register which holds the number of strokes in the image. Its contents need be loaded once at the time that the initial image size is set. Its contents are copied to a down counter at the time of a print request. With each stroke packet sent, the counter is decremented. It is this counter decrementing to zero that completes the image print process and results in the image complete interrupt.

10

5. Head number register - this is an 8 bit register which holds the head number which is inserted into the packet header to be read by the print engine. Normally, this register need only be written once after power turn on, the head number never changing for the associated print engine port.

15

6. Product index number register - this is an 8 bit register which holds the product index value which is inserted into the packet header to be read by the print engine. This number is intended to be an aid in identifying which print image had any detected problem associated with it, whether it be a communication error, print error or another different error when reporting back to the image processor.

20

7. Message tagbit register - this is a 1 bit register which is set by the image processor if a particular image is to be responded to by the print engine with a report

25

back to the image processor 20 through the DDI channel upon its receipt. The register is automatically cleared at the end of the first stroke packet. It must be specifically set for each print image needing the report back. This choice will be based on software for the application and the degree of certainty of good print needed for that application.

8. Internal Product Store register - this is a 16 bit counter which stores the product detect repeat interval, counted in divided encoder pulses, i.e. stroke increments. The same is true if internal encoder is used here.

Several hardware parameters must be stored in order to control the printing process and the stroke manager 72. These parameters and explanations of each are listed below. Some parameters are accessed collectively as individual bits of one of the Configuration registers. Other parameters are separately addressed for each bit setting or value stored. This simplifies certain software routines that need quick access to the parameter and without the risk or delay associated with a multiple parameter word access. Most of the multiple parameter configuration registers are one time set parameters.

The parameters are defined for a two head controller. In such a printer, there is the possibility of two separate product detectors 26 and two separate shaft encoders 24. Then each head can be programmed to be associated with either detector and either encoder. However, in the above example, there is only one head 12, so only one product detector 26 and one encoder 24 are used. Also, some of the selection bits become unused, since there is no choice of which encoder and which product detector to associate with the head.



## PARAMETERS

## pd\_encoder\_selectffl -

Used to select which of the two main shaft encoder inputs are associated with Product  
5 Detect #1. In other words, this bit assigns the shaft encoder to the particular product  
detector. Setting this to the Bit = 0 selects encoder 1; otherwise encoder 2 is selected.

## SLI\_baudrate1(1-0) -

These two bits select one of four Stroke Level Interface serial stroke data baud rates for  
10 print engine #1 (head #1). The choices are:

	00	4MHz
	01	8MHz
	10	16MHz
15	11	32MHz

## autoencode\_mode\_ffl -

When auto-encoding is used, this bit controls whether a single product detector 26 is used  
or timing between the two product detector pulses PD1 and PD2 is used to set the  
20 internally timed stroke rate for that particular print message. Bit = 0 enables timing of  
the duration of PD 1 true to be measured as the auto-encode measurement time. When  
PD 1 has returned false, this count allows software to determine the internal stroke rate to  
use for the next product mark. The product detect must also be set to trailing edge trigger  
in order to use the new time on the same product. The counter measures ticks of the 8  
25 MHZ clock.

When Bit = 1, the autoencode timing begins with the lead edge of PD 1 and ends with the lead edge of PD 2.

pd\_selectffl

- 5           Selects which product detector 26 and the assigned shaft encoder 24 is associated with Head #1. Bit = 0 assigns PD1 to head 1. Otherwise, PD 2 is assigned to head 1.

pd\_invffl

- 10           Selects whether the PD1 signal is inverted or not inverted. Bit = 0 does not invert the PD signal; bit = 1 causes the signal polarity to be inverted. This choice is made during installation of the product detector 26 and normally not changed.

pd\_lead\_edge\_selffl

- 15           Selects whether the lead or trail edge of the product detect signal triggers the detect event. Bit = 1 selects the lead edge of the product detect signal; bit = 0 selects the trail edge.

reverse\_printffl

- 20           Controls the direction of the scanning of the stroke address table in memory by the stroke manager hardware. For reverse printing, i.e. printing which starts at the end of the message instead of the beginning, this bit is set high. The software must have written the last address of the table as the starting address. Then the hardware will retrieve successive table addresses by decrementing the table address counter, rather than incrementing.

print\_enableffl

Enables printing for print head #1. Software should only set this bit true after all hardware has been configured and the Stroke Manager 72 is set up.

#### quadrature\_enableff\_main1

- 5           Used to activate quadrature shaft encoder logic which then detects each quarter cycle of a two channel shaft encoder signal. This also allows hardware to sense forward and backward motion, to detect backlash motion, and to inhibit erroneous backward trigger pulses, only commencing printing once the travel position has returned to the latest progress point.

10

#### reverse\_dirff\_main1

Sets the “command direction” for encoder travel. Some applications may involve bidirectional printing and thus requiring alternating the transport motion direction. The quadrature encoder logic must be told which direction is to be printed.

15

#### stroke\_invertff1

Controls whether print strokes are to be printed normally or inverted.

#### enc\_dir\_invff\_main1

- 20           Defines the quadrature encoder waveform direction that is to be treated as the print direction for “forward motion”. Bit = 0 is interpreted as forward motion causing encoder channel A to lead encoder channel B.

#### str\_source\_selfff1

Used to select whether an external shaft encoder 24 or an internal timer is used to trigger stroke printing. Bit = 0 selects internal timer for stroke pulses; bit = 1 selects the shaft encoder associated with the head.

5    internal\_pd\_selectff1

Used to select internal product detector 26 instead of external. When bit = 1, selects internal timing generator for product detect pulses for head 1.

pd\_encoder\_selectff2

10        Used to select which of the two main shaft encoder 24 inputs are associated with Product Detector #2. In other words, this bit assigns the shaft encoder to the particular product detector. Bit = 0 selects encoder 1; otherwise encoder 2 is selected.

SLI\_baudrate2(1-0)

15        These two bits select one of four Stroke Level Interface serial stroke data baud rates for print engine #2 (head #2). The choices are:

	00	4MHz
	01	8MHz
20	10	16MHz
	11	32MHz

autoencode\_mode\_ff2

Will be used to activate the selected Auto-encode mode. Not implemented at present.

Currently, there is no choice for Product Detector 2. If auto-encoding is used, it can only be based on the timing of Product Detector 2.

5     pd\_selectff2

Selects which product detector and the assigned shaft encoder is associated with Head #2.

Bit = 0 assigns Product Detector 1 to head 1. Otherwise, Product Detector 2 is assigned to head 1.

10    pd\_invff2

Selects whether the Product Detector 2 signal is inverted or not inverted. Bit = 0 does not invert the Product Detect signal; bit = 1 causes the signal polarity to be inverted. This choice is made during installation of the product detector and normally not changed.

15    pd\_lead\_edge\_selff2

Selects whether the lead or trail edge of the product detect signal triggers the detect event.. Bit = 1 selects the lead edge of the product detect signal; bit = 0 selects the trail edge.

20    reverse\_printff2

Controls the direction of the scanning of the stroke address table in memory by the stroke manager hardware. For reverse printing, i.e. printing which starts at the end of the message instead of the beginning, this bit is set high. The software must have written the last address of the table as the starting address. Then the hardware will retrieve successive table addresses by decrementing the table address counter, rather than incrementing.

25

#### print\_enableff2

Enables printing for print head #2. Software should only set this bit true after all hardware has been configured and the Stroke Manager is set up.

5

#### quadrature\_enableff\_main2

Used to activate quadrature shaft encoder logic which then detects each quarter cycle of a two channel shaft encoder signal. This also allows hardware to sense forward and backward motion, to detect backlash motion, and to inhibit erroneous backward trigger pulses, only commencing printing once the travel position has returned to the latest progress point.

10

#### reverse\_dirff\_main2

Sets the “command direction” for encoder travel. Some applications may involve bidirectional printing and thus requiring alternating the transport motion direction. The quadrature encoder logic must be told which direction is to be printed.

15

#### stroke\_invertff2

Controls whether print strokes are to be printed normally or inverted.

20

#### enc\_dir\_invff\_main2

Defines the quadrature encoder waveform direction that is to be treated as the print direction for “forward motion”. Bit = 0 is interpreted as forward motion causing encoder channel A to lead encoder channel B.

25

str\_source\_selff2

Used to select whether an external shaft encoder or an internal timer is used to trigger stroke printing. Bit = 0 selects internal timer for stroke pulses; bit = 1 selects the shaft encoder associated with the head.

5

internal\_pd\_selectff2

Used to select internal product detector 26 instead of external. When bit = 1, selects internal timing generator for product detect pulses for head 2.

10

The shaft encoder logic, resident in controller hardware 14, consists of several hardware functions. These include:

- 1) The selection of which encoder 24 to assign to which product detector (when appropriate). Uses a configuration register bit.
- 15 2) Selection of whether or not quadrature encoder signals are used. Uses a configuration register bit.
- 3) Direction select for forward for quadrature encoders and direction sensing. Uses a configuration register bit.
- 4) Backlash counter logic - up to 63,488 shaft encoder increments in the reverse  
20 direction as compared to the selected "forward" direction can be counted. This logic will suppress any encoder ticks until all backward motion has been recovered up to this maximum.
- 5) Speed sensing for external shaft encoders - a register will store a count of the number of undivided encoder ticks that occur each 0.1 seconds. This register is  
25 also updated each 0.1 seconds.

6) Pulse rate divide of encoder pulses used to trigger stroke printing - the raw or undivided encoder pulse rate may be divided by any integer or non-integer value that is expressable as a ratio of two eight bit numbers. For example, a divide by 4 can be expressed as a ratio of 4 to 1, 16 to 4, 128 to 32, etc. A divide by 4.75 can  
5 be expressed as a ratio of 19 to 4, 57 to 12, 152 to 32, etc. In each case, the two eight bit values are stored in a single 16 bit register, the numerator being the upper eight bits and the denominator being the lower eight bits.

7) Providing encoder pulses (fixed divide by 100) to the bulkhead board processor.  
10

The interrupt logic is structured to drive the two external interrupt inputs of the Motorola 68322 image processor 20 based on all sources of interrupts in the controller hardware. The print related interrupts are directed to interrupt 0. All other interrupts are directed to interrupt 1 of the processor. Interrupt priority logic is used to choose which of several simultaneous inputs  
15 causes the actual interrupt. An interrupt "vector" may be read by the processor to determine which of the several possible interrupt inputs has caused the interrupt.

For all of these interrupts, a set of logic is defined with similar characteristics. The interrupt is enabled or disabled by software writing a bit value to a specific address. Once  
20 enabled, an interrupt input active edge causes an interrupt to be set and an interrupt flag to be set. Both the interrupt and the flag must be cleared before another interrupt from that source may occur. The hardware is arranged so that clearing the flag also clears the interrupt. However, for cases where the reoccurrence needs to be suppressed even though the interrupt is cleared, the flag is left set. This will suppress further interrupts until the flag is cleared. In addition, the  
25 occurrence of an interrupt input before the flag has been cleared is defined as an over speed



condition. That over speed is registered for some of the interrupts and can be read by the processor as a fault condition.

## INTERRUPTS

5

### PROCESSOR INTERRUPT 0

Interrupt 0 is mapped to the six interrupt sources listed here. Each interrupt is enabled with a separate write to a unique address. The product detect and print request interrupts also have fault registers attached. Each of these will register a fault if a second interrupt condition occurs before the flag has been cleared. The product detect logic also has a bounce fault register which is set if multiple product detect transitions occur before the interrupt itself is cleared.

Intr0\_vector(5 downto 0)

15

5. Product Detect 1

4. Product Detect 2

3. Print Request 1

2. Print Request 2

20

1. Image Done 1

0. Image Done 2

### PROCESSOR INTERRUPT 1

Interrupt 1 is mapped to the seven sources listed here. All interrupts are enabled by writing specific bits to the same address, a general interrupt enable address.

Intr1\_vector(8 downto 0)

5           8, 7, 6, 3, 2, 1, 0 used; 5, 4 unused

8. IO Processor Comm.

7. Merge Data Serial Port

6. Dual Port RAM

10          3. DDI 1

2. DDI 2

1. Parallel Port (normally not connected)

0. Spare (unused)

15           Other types of print engines, besides stroke based engines can also be supported in the system of the invention using the same physical interface as has been described hereinbefore. The SLIPP interface is a data packet communication channel. The form and definition assigned to the packet contents can be altered. The stroke trigger signal can also be redefined. Two examples of such a system would be vector based print engine and a bitmap based print engine  
20           which may print the image, in effect all at once, with one trigger command.

            In the case of a vector based engine, the entire list of vectors would be sent to the print engine. The entire list may be needed before printing starts. Assuming a moving substrate application, the print delay and print queuing logic still applies. The image processor still builds  
25           the images, but vectors rather than bitmaps would be used, and queued up as product detect

signals occur. The trigger signal would still be useful where the substrate is moving because it would allow the print engine to adjust its vector trajectory to keep aligned with the moving substrate. The trigger pulses would represent a predefined substrate pitch and therefore velocity. Even if the substrate is stationary, the trigger pulse could represent the print trigger command.

5

In the case of a single command, full bitmap transfer print mechanism, again the full bitmap image would need to be transferred over the SLIPP port prior to printing. The trigger signal would be defined as the print trigger signal.

CLAIMS:

1. A control system capable of being connected to a plurality of printers (12, 30), said printers (12, 30) employing print engines (30) of the same or different configurations, and using the same or different data protocols, each printer (12, 30) including a print head (12) controlled  
5 by the associated print engine (30) to mark a substrate, at least one of said print head (12) and said substrate moving relative to the other at a rate which is not controlled by said control system or said printer (12, 30), said control system comprising: a) an image processor (20) capable of generating image data in more than one protocol representing individual pixels or vectors to be marked on said substrate; b) a synchronization signal generator (24, 72) for generating  
10 synchronization signals representative of the relative location of said print head (12) and said substrate; and c) an interface (40) connecting said image processor (20) and said synchronization signal generator (24, 72) to said print engine (30), and allowing said image data and said synchronization signals to be transferred to said print engine (30), said interface (40) accommodating different print engine protocols, thereby to allow various print engines (30) to be  
15 interfaced to the same control system.
2. A control system in accordance with Claim 1 wherein said synchronization signal comprises a stroke signal and wherein said synchronization signal generator (24, 72) comprises: a shaft encoder (24) for producing a periodic signal with a frequency representative of the rate of motion of said substrate; and a stroke manager (72) for producing stroke pulses at a frequency  
20 substantially proportional to said frequency.
3. A control system according to Claim 1 or Claim 2 further comprising memory (73) for holding said imaging data, said synchronization signal generator (24, 72) triggering output of said imaging data from said memory (73) to said print engine (30).
4. A control system in accordance with Claim 3 wherein said synchronization signal  
25 generator (24, 72) uses direct memory access (DMA) to output said image data.

5. A control system in accordance with Claim 3 or Claim 4 wherein image data is retained in said memory (73) between different print jobs containing identical elements of image data, thereby allowing re-use of image data.

6. A control system in accordance with any one of the preceding claims wherein said interface (40) further permits print engine control signals from said control system (20, 24, 72, 40) to said print engine (30) to allow configuration of said print engine (30).

7. A control system in accordance with Claim 1 wherein the synchronization signals are generated at a required rate in a printer (12, 30), in response to pulse signals from a shaft encoder (24) at a shaft encoder pulse rate, said synchronization signals being generated by: 1) providing first and second integer values which, when said first integer value is divided by said second integer value provides a desired ratio of the shaft encoder pulse rate to the required rate of stroke pulses; and 2) using said integer values to generate synchronization pulses substantially at said shaft encoder pulse rate divided by said ratio.

8. A control system in accordance with Claim 7 wherein said synchronization signals are output at intervals which are substantially multiples of the shaft encoder interval, but wherein consecutive synchronization pulse intervals differ in the number of multiples of the shaft encoder interval employed, thereby permitting the average output synchronization pulse rate to match the required rate of synchronization pulses.

9. A control system for a printer (12, 30) having a print engine (30) and a print head (12) controlled by said print engine (30) for marking a moving substrate, said control system comprising: a) an image processor (20) for generating images to be output to said print engine (30), for causing said print head (12) to mark said substrate; b) a counter (62) operating independently of said image processor (20) and responsive to signals generated by a shaft encoder (24) associated with movement of said substrate; c) means for reading the value of said counter (62), adding a predetermined value to said counter value and storing the result as a

compare value in response to the detection of said substrate; and d) comparing means (71) operating independently of said image processor (20) for initiating printing of said images on said substrate when the value of said counter (62) matches said compare value.

10. A control system in accordance with Claim 9 wherein said image processor (20) comprises said means for reading, and wherein an interrupt routine is triggered on said image processor (20) when said substrate is detected, said interrupt routine including the steps of reading said value from said counter (62), adding said predetermined value to said counter value and storing the result as said compare value.

11. A control system in accordance with Claim 9 or Claim 10 wherein said counter value is latched into a capture register (64) and the capture register value is read by said means for reading.

12. A control system in accordance with Claim 9 or Claim 10 or Claim 11 wherein said compare value is stored in a register (66).

13. A control system in accordance with any one of Claims 9 to 12 further comprising a stroke manager (72) for generating stroke signals used by said print engine (30), said stroke manager (72) receiving signals from said shaft encoder (24) from which the stroke manager (72) generates stroke signals, said stroke manager (72) being arranged to receive a signal from said comparing means (71) when said counter and compare values coincide and thereupon commence generating stroke signals.

14. An interface for connecting a control system to one of a plurality of printers (12, 30), said printers (12, 30) employing print engines (30) of the same or different configurations, and using the same or different data protocols, each printer (12, 30) including a print head (12) controlled by the associated print engine (30) to mark a substrate, at least one of said print head (12) and said substrate moving relative to the other at a rate which is not controlled by said control system or said printer (12, 30), said interface comprising: a) a stroke signal interface (42) capable of

transferring stroke data in more than one protocol representing pixels in said stroke from said control system to said print engine (30); and b) a synchronization signal interface (42) for transferring synchronization data representing the position of said substrate relative to said print head (12) from said control system to said print engine (30).

- 5 15. An interface in accordance with Claim 14 comprising a parallel to serial convertor (422) located in association with said control system for converting said stroke data into serial format, and a serial to parallel convertor (422) located in association with said print engine (30) for converting said stroke data into parallel format.

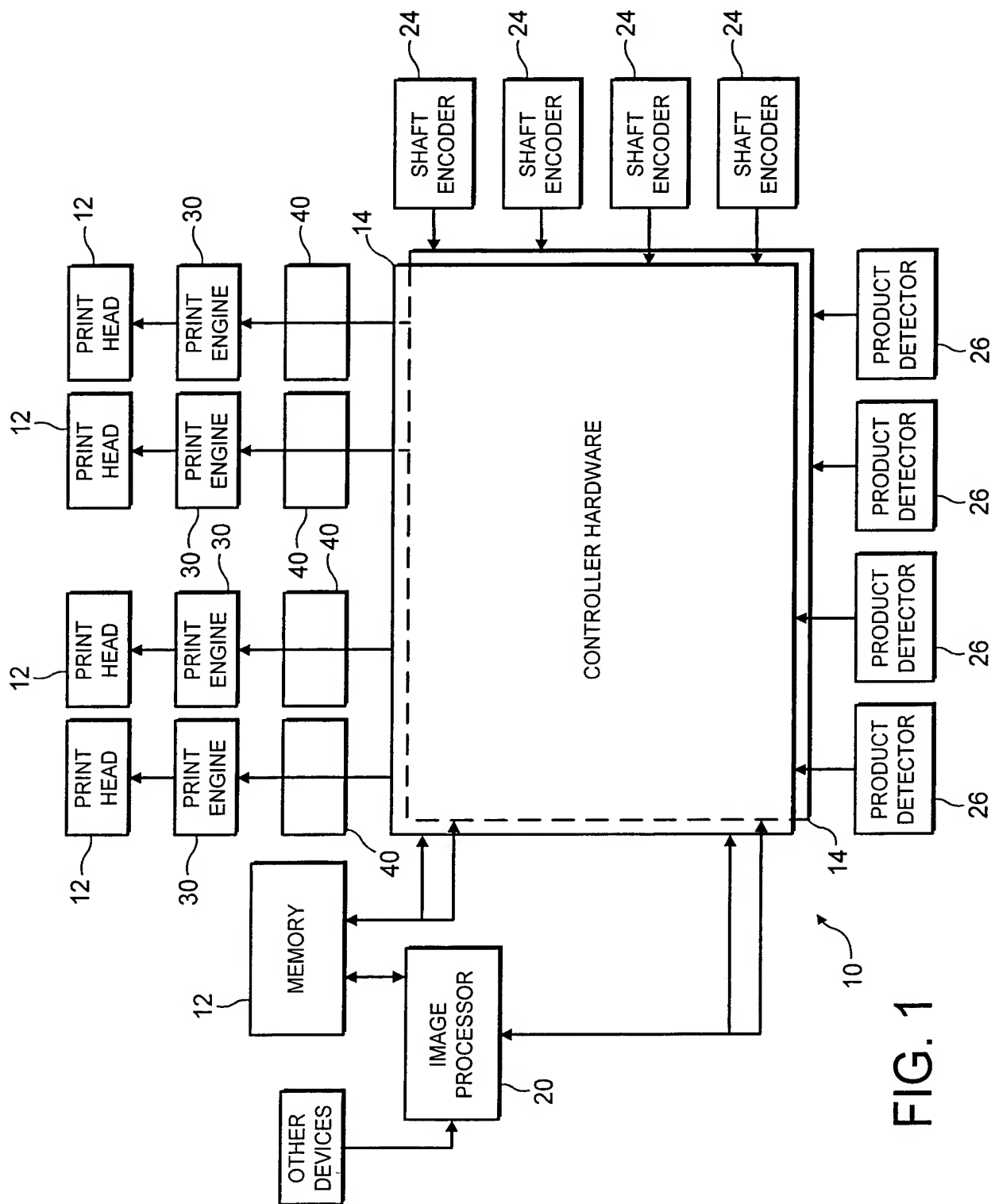


FIG. 1



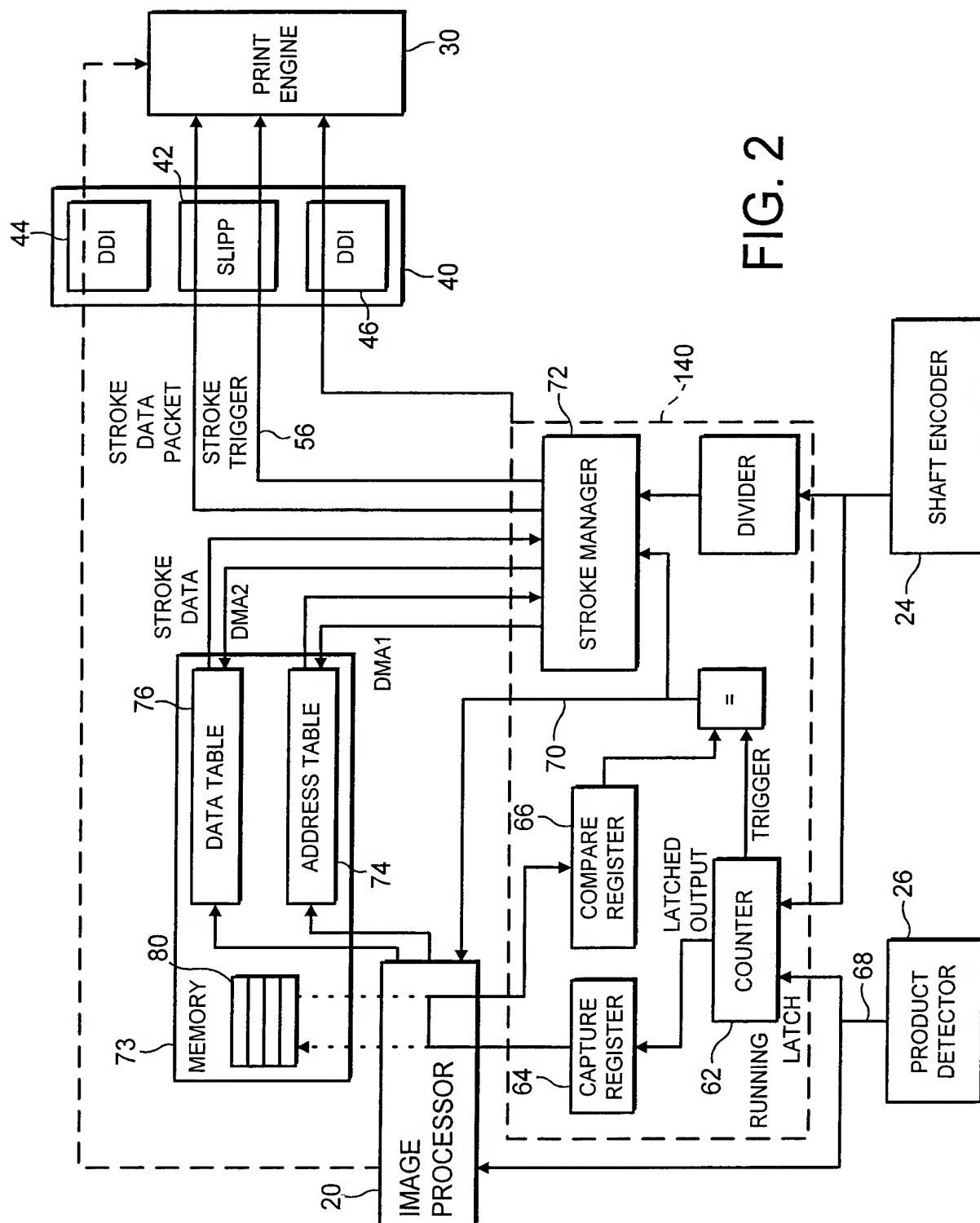


FIG. 2

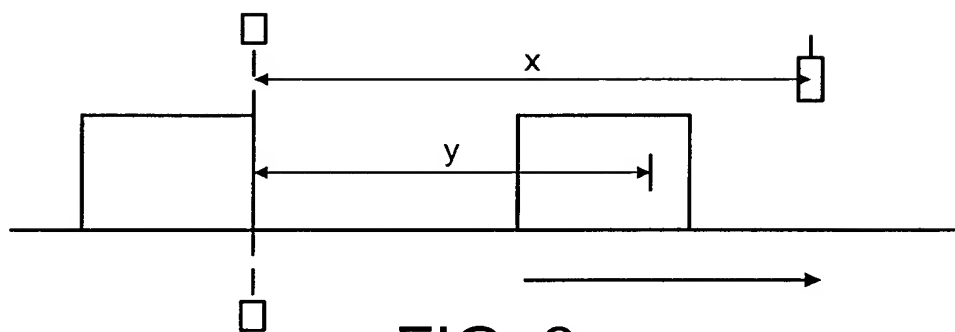


FIG. 3

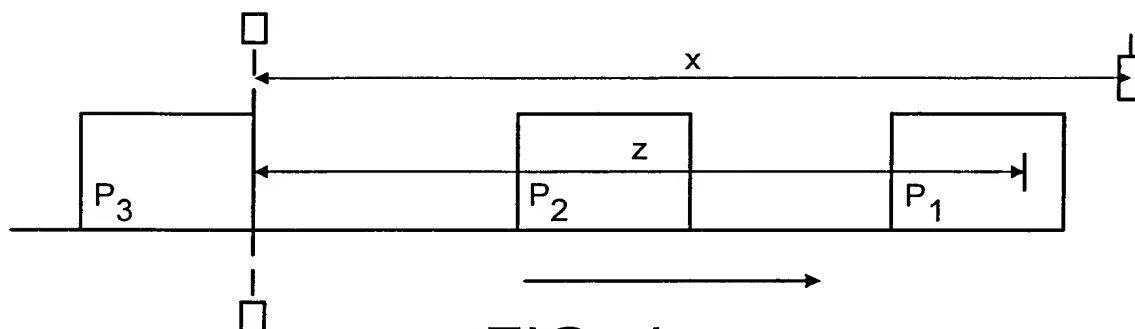


FIG. 4

4 / 9

ENCODER PULSE #	COUNTER VALUE	STROKE PULSE
1	2	
2	4	
3	6	
4	8	
5	10	
6	12	
7	14	
8	16->1	YES
9	3	
10	5	
11	7	
12	9	
13	11	
14	13	
15	15->0	YES
16	2	
17	4	

FIG. 5

Mr. John Smith > Head 1  
2500 Rodeo Lane > Image  
Anywhere, CA 12345 > Region

-----  
Mr. Smith: PLease see or Advertisement on > Head 1  
Page 14 for special prices. > Image  
> Region

FIG. 6

\_\_\_\_ HEAD 1 IMAGE \_\_\_\_ (UPPER 2/3 OF LARGE CHARS.)  
**X Y Z**  
-----  
PN345987 >HEAD S IMAGE  
Exp 6apr2002 > HEAD 2

FIG. 7

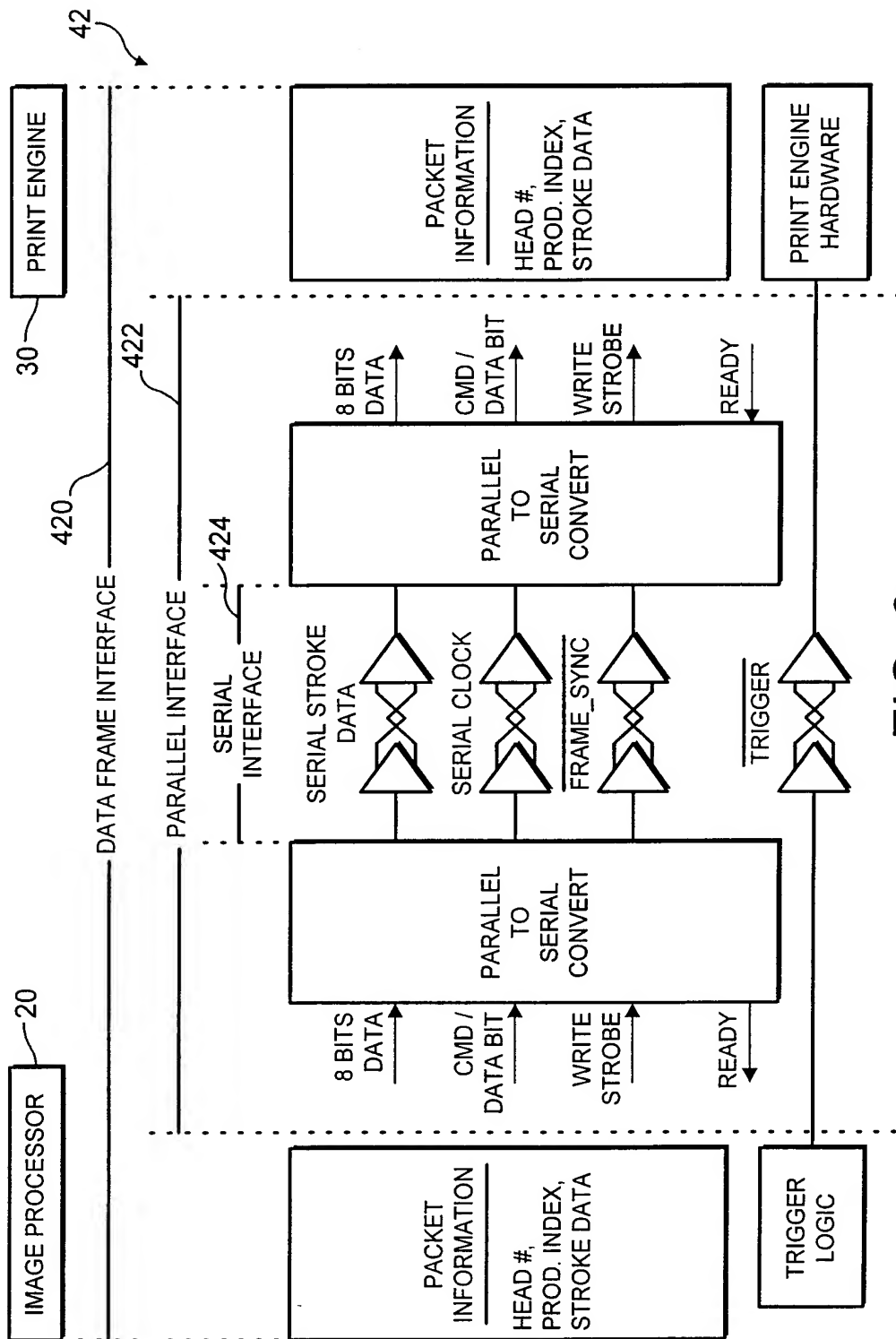
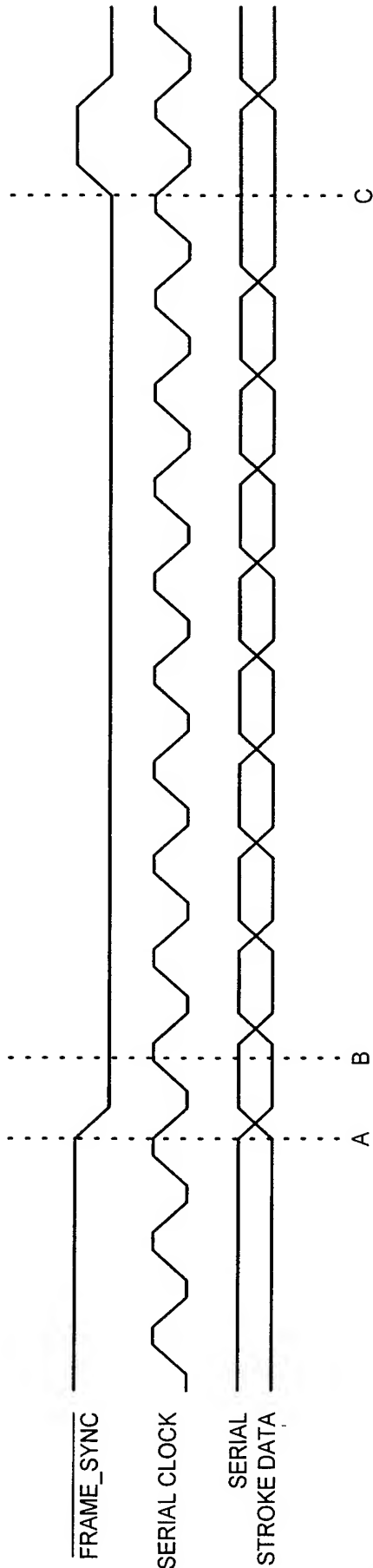


FIG. 8

PACKET WORDS	FIBER CHANNEL S.C. Byte Name (S.C. Code Name)	Cmd/*d 1 = Command 0 = data	BIT PATTERN (MSB - LSB)
SOF_F, Stroke Data Packet 1st Stroke OR	K28.1 (C01)	1	T000 0001
SOF_N, Stroke Data Packet Next Stroke OR	K28.3 (C03)		0000 0011
SOF_L, Stroke Data Packet Last Stroke	K28.2 (C02)		0000 0010
Printhead number (1 to n)	NA	0	Binary
Product Index (0 to 255)	NA	0	Binary
STROKE PATTERN DATA  Number of bytes and format vary for each print technology and can include special bytes before the data for Print Technology Specific usage.	NA	0  0  0  0	Byte order -  bottom of stroke to top, lowest byte 1st. Bit order - least significant at bottom
EOF, End of Frame delimiter	K27.7 (C09)	1	0000 1001

FIG. 9



- A- FRAME SYNC IS MADE TRUE (HIGH) ON THE FALLING EDGE OF SERIAL CLOCK CONCURRENTLY. THE FIRST BIT OF DATA IS PLACED ON THE SERIAL OUTPUT. DATA IS SHIFTED LOW-BIT FIRST.
- B- THE RECEIVER SHOULD SAMPLE DATA ON THE RISING EDGE OF SERIAL CLOCK. THIS CLOCK IS PROCESSED THROUGH THE PLD ALONG WITH THE FRAME SYNC AND DATA SO THAT IT'S SKEW, WITH RESPECT TO THE INPUT CLOCK, IS SIMILAR. THIS SHOULD MINIMIZE THEIR SKEW WITH RESPECT TO EACH OTHER. IT ALSO ALLOWS THE PLD TO INCLUDE A PROGRAMMABLE CLOCK DIVIDER TO REDUCE SMI FOR APPLICATIONS WHICH DO NOT REQUIRE THE FULL 38 MMS SPEED.
- C- FRAME SYNC IS MADE FALSE (LOW) ON THE FALLING EDGE OF SERIAL CLOCK AFTER THE PARITY BIT HAS BEEN SHIFTED OUT. IT WILL REMAIN LOW FOR A MINIMUM OF ONE BIT-TIME TO ALLOW THE RECEIVER TO MOVE THE CAPTURED DATA TO IT'S FIFO OR PARALLEL BUFFER. IT'S MAXIMUM LOW TIME WILL BE DETERMINED BY THE AVAILABILITY OF DATA AT THE TRANSMITTER - IT WON'T BE MADE TRUE AGAIN UNTIL ANOTHER PARALLEL BYTE IS WRITTEN.

FIG. 10

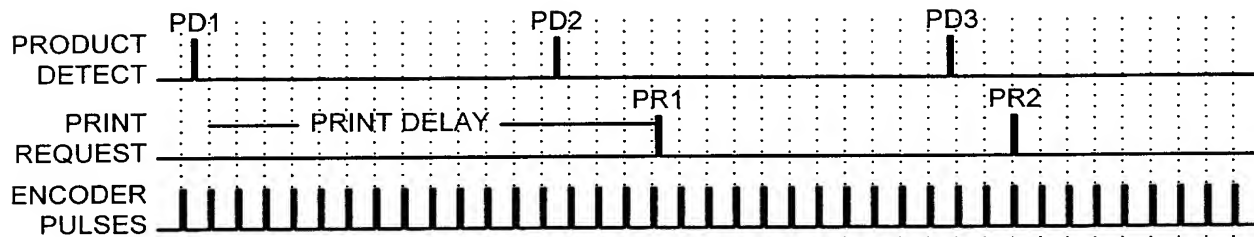


FIG. 11

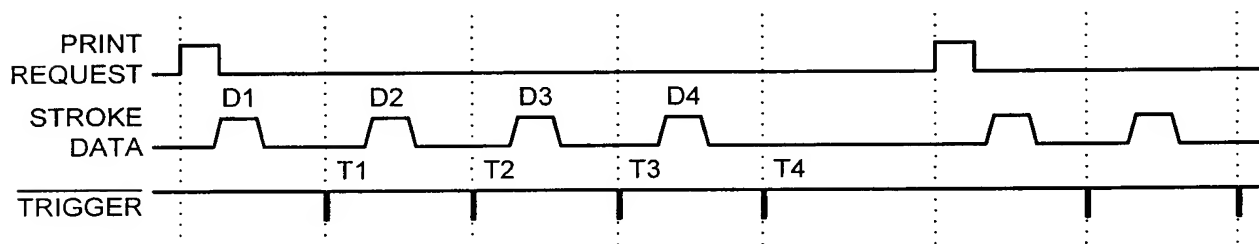


FIG. 12

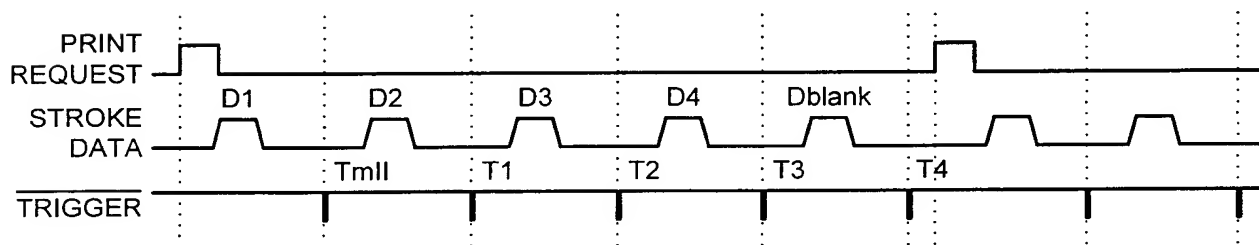


FIG. 13

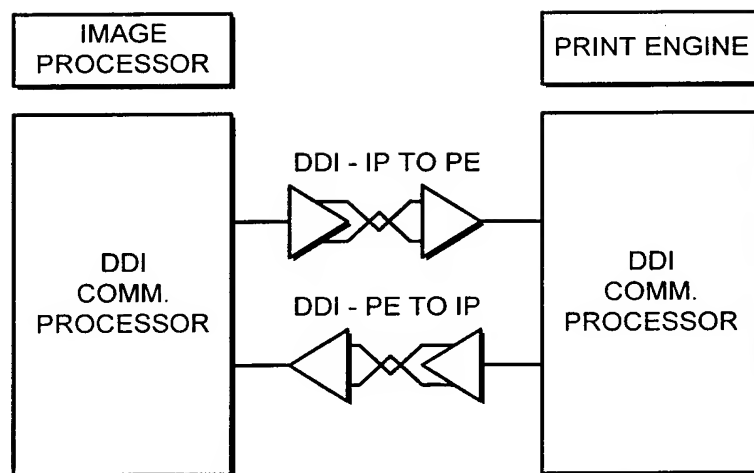


FIG. 14

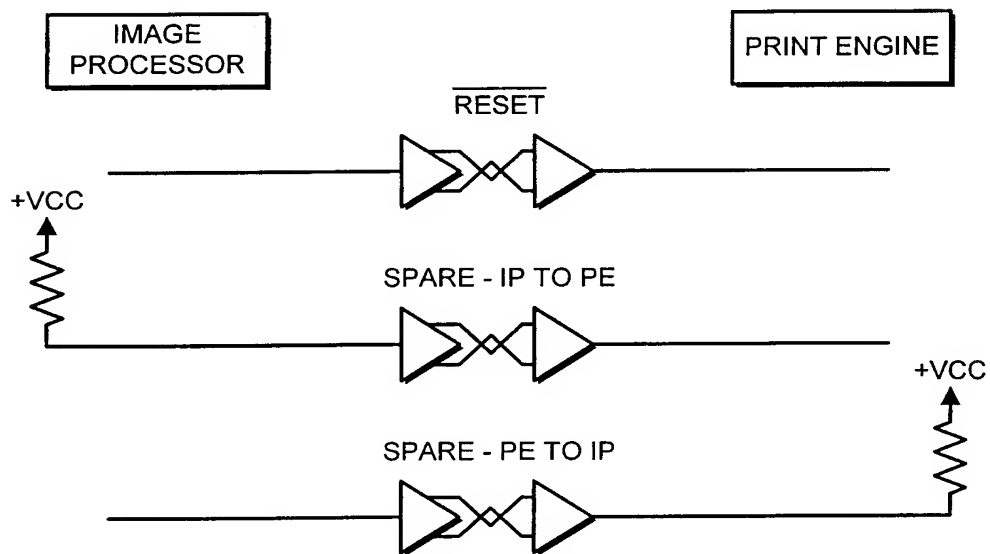


FIG. 15



# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/GB 00/03173

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 7 G06K15/16 G03G15/00

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
IPC 7 G06K G03G B07C B41J

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US 4 627 222 A (CANTILE JACK W) 9 December 1986 (1986-12-09) column 4, line 30 -column 5, line 58; figures 1,12,13 column 8, line 58 -column 9, line 27; claims 1-4 -----	1-3,5,6, 14,15 4,7-13
X Y	US 3 911 818 A (MACILVAINE DONALD A) 14 October 1975 (1975-10-14) column 7, line 33 -column 8, line 11; figures 1,3 -----	14,15 4,7-13
A	US 5 439 340 A (VOLKMANN JEFFREY R) 8 August 1995 (1995-08-08) claims 15,16 -----	1,6,15

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

\* Special categories of cited documents:

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

- \*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- \*&\* document member of the same patent family

Date of the actual completion of the international search

24 November 2000

Date of mailing of the international search report

01/12/2000

Name and mailing address of the ISA  
European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Cardigos dos Reis, F

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/GB 00/03173

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 4627222 A	09-12-1986	CA 1259273 A US 4739606 A	12-09-1989 26-04-1988
US 3911818 A	14-10-1975	AR 207334 A BE 819313 A CA 996176 A DE 2439123 A FR 2242248 A GB 1475937 A IE 39997 B IT 1018289 B JP 1367091 C JP 50074342 A JP 61027192 B NL 7411022 A,B, SE 402360 B SE 7410808 A ZA 7405169 A	30-09-1976 16-12-1974 31-08-1976 06-03-1975 28-03-1975 10-06-1977 14-02-1979 30-09-1977 26-02-1987 19-06-1975 24-06-1986 06-03-1975 26-06-1978 05-03-1975 27-08-1975
US 5439340 A	08-08-1995	NONE	